

Cortex-M sucht Anschluss

ARM sieht für seine aktuelle Mikrocontroller-Serie einen neuen Debug-Anschluss vor.

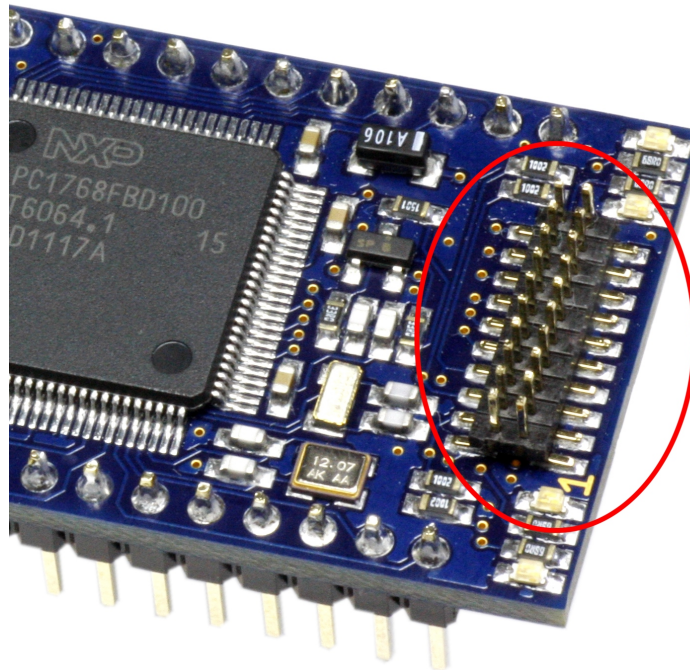


Abbildung 1: Controllerboard mit Cortex-M3 und Cortex-Debug+ETM Interface

1 Die Debug-Schnittstelle beim ARM Cortex-M

Zur Erläuterung der Hintergründe dient uns zunächst ein Blick auf die innere Struktur des Debug-Ports im Mikrocontroller.

Das Schema in Abbildung 2 verdeutlicht die von ARM verwendeten Begrifflichkeiten: Es wird zwischen Debug- und Trace-Port-Interface unterschieden. Das Debug Interface kann wahlweise JTAG oder Serial Wire oder beides gemeinsam unterstützen und bietet den Zugriff auf die grundlegenden Funktionen wie Flash Programmierung, Start-, Stopp-, Einzelschrittkontrolle sowie das Setzen von Breakpoints.

Die Zusatzfunktionen wie Instruction und Instrumentation Trace sowie DWT werden durch das Trace Port Interface weiterverarbeitet. Dieses kann als seriell arbeitendes Serial Wire oder als getaktetes, 4-Bit breites Parallelinterface implementiert sein. Letzteres ermöglicht es, die vom Real-Time Instruction Trace Modul (ETM - Embedded Trace Macrocell) erzeugten Daten an ein Debug-Werkzeug weiterzuleiten.

Zu erkennen ist ebenfalls, dass die Daten vom Trace Port und Debug Interface über einen Multiplexer namens SWJ zusammengeführt werden. SWJ ist die Abkürzung für Serial Wire JTAG und deutet auf die Unterstützung beider Debug-Standards hin. Am Schema klärt sich auch die Existenz von zwei möglichen Steckverbindern: ein 10-poliger wird für Controller ohne ETM, also ohne Real-Time-Trace mit seinem 4-Bit Parallelinterface verwendet. Für Controller mit Vollausstattung findet der 20-polige Stecker Anwendung.

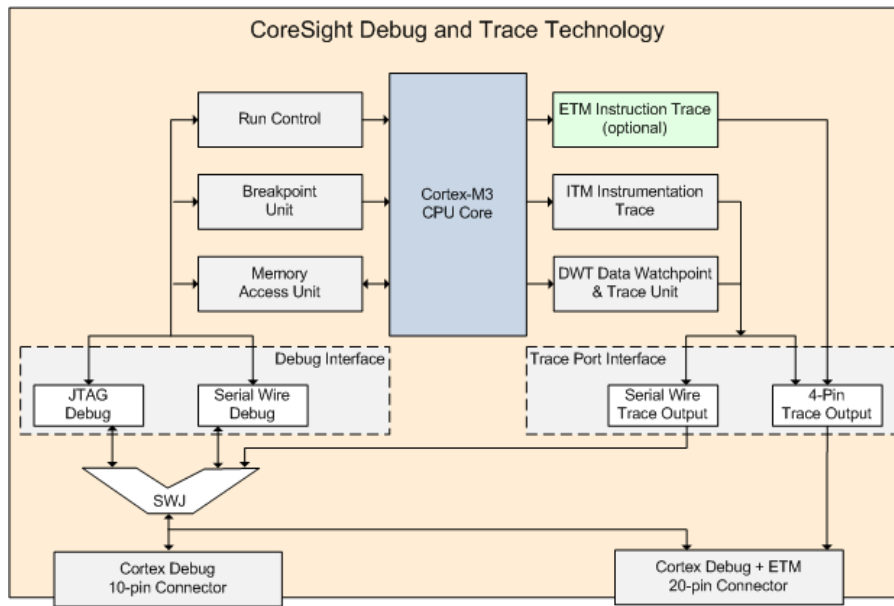


Abbildung 2: Struktureller Aufbau des Debug+Trace-Ports [1]

1.1 ARM-JTAG

Auf Boards mit aktuellen Cortex-M Controllern sollte vorrangig der Cortex-Debug-Stecker eingesetzt werden. Jedoch finden sich auch Boards mit klassischem ARM-JTAG Steckverbinder, dessen Pinbelegung auf dem Schaltplan (3) dargestellt ist. Durch die Mehrfachnutzung der Signalleitungen kann über den ARM-JTAG Steckverbinder auch SWD und SWV genutzt werden. Pin 2 ermöglicht die Spannungsversorgung des Debug-Adapters vom angeschlossenen Zielsystem. Im hier erläuterten Beispiel muss dazu der Jumper kurzgeschlossen werden.

1.2 Das Serial Wire Protokoll

Serial Wire Debug (SWD) ist die bevorzugte Interface-Art beim Cortex-M. In Minimalausstattung benötigt die Debug-Verbindung damit nur 3 Leitungen - Die Zeitbasis (SWDCLK), das bidirektionale Datensignal (SWDIO) sowie die Bezugsmasse (GND).

1.3 Serial Wire Viewer

Mit der optionalen Serial Wire Viewer (SWV) Komponente wird ein serielles Verfahren beschrieben, welches vom Controller erzeugte Debug-Meldungen an den Debug Adapter überträgt. Diese Übertragung erfolgt über das Signal SWO (Serial Wire Out). Im Schema 2 bildet ein langer Pfeil vom Serial Wire Trace Output zum SWJ-Multiplexer diese Verbindung ab.

1.4 Instruction Trace

Einige Cortex-M Chips beinhalten das ETM-Modul (Embedded Trace Macrocell), welches dem Trace-Adapter detaillierte Einblicke in den Programmablauf ermöglicht. Um auch bei hohem Prozessortakt und hoher Systemauslastung verlässlich die anfallenden Daten weiterzureichen, werden die anfallenden Daten wie beschrieben parallel mit einer Breite von 4 Bit übertragen. Diese Parallelanbindung benötigt fünf zusätzliche Anschlusspins (vier Datensignale und ein

Taktsignal), was die Verwendung eines größeren, in diesem Falle 20-poligen Steckverbinders notwendig macht.

2 Die Ausführung der Debug-Schnittstelle

2.1 Steckverbinder

ARM empfiehlt für den Einsatz auf Cortex-M basierten Boards die Verwendung von Stifleisten im 1,27mm-Raster. Um Platz auf der Leiterplatte zu sparen, lässt sich somit die gleiche Anzahl Signale auf halber Steckverbinderbreite herausführen. ARM unterscheidet zwischen zwei Steckerformen: Cortex Debug als 2x5-poliger Anschluss und Cortex Debug+ETM als 2x10-poliger Steckverbinder. Dabei ist die erste Hälfte dieser beiden Anschlüsse identisch - Cortex Debug+ETM stellt lediglich zusätzlich die Trace-Signale zur Verfügung. Bei beiden Varianten ist Pin 7 „keyed“; dort befindet sich steckerseitig kein Stift und buchsenseitig keine Öffnung.

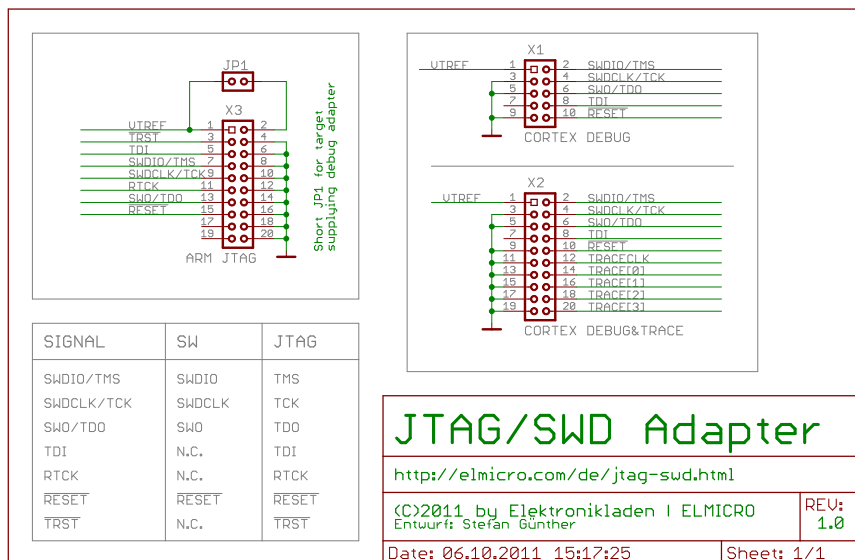


Abbildung 3: Schaltplan eines JTAG/SWD-Adapters

2.2 Signalbelegung

Aus Schema 2 ist bekannt, dass das Debug Interface eines Cortex-M entweder JTAG, SWD oder beides unterstützen kann. In jedem Fall gelangen die Signale dieser Interfaces zunächst in den Multiplexer namens SWJ und werden dann am 10-poligen Cortex-Debug-Stecker (oder am „unteren“ Teil des 20-poligen Cortex-Debug+ETM Anschlusses) bereitgestellt.

Da stets nur entweder JTAG oder SWD für die Kommunikation genutzt werden kann, hat sich ARM für die Doppelbelegung der JTAG-Signale entschieden. Somit teilt sich beispielsweise JTAG-TMS eine Signalleitung mit SWDIO. Die Zuordnung aller Signale findet sich im Schaltplan zum JTAG/SWD-Adapter (Abbildung 3).

3 Anschluss für alle

Leider ist alle Theorie grau und so finden sich neben vorbildlichen Produkten auf Cortex-M Basis mit Cortex-Debug oder Debug+Trace eben auch solche mit traditionellem ARM JTAG Anschluss. Die derzeit verfügbaren Debug-Adapter weisen ebenfalls teilweise Buchsen im 2,54mm-Raster und teilweise im kleineren, 1,27mm-Raster auf. Desweiteren ist es schlicht unmöglich, eine Anschlussbuchse, die für den 2x5-poligen Cortex-Debug Stecker vorgesehen ist, auf den 2x10-poligen Cortex Debug+ETM Anschluss zu stecken.

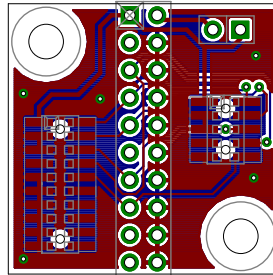


Abbildung 4: Layout eines JTAG/SWD-Adapters

Um mit einem einzigen Debug-Tool flexibel auf alle Targets zugreifen zu können, dürfte sich ein kleiner Adapter als nützlich erweisen. Ein solcher Universaladapter könnte beispielsweise aussehen wie in Abbildung 4. Die Eagle-Dateien für diesen Adapter stehen zum kostenfreien Download für Interessierte bereit [2].

4 Quellennachweise

- [1] http://www.keil.com/support/man/docs/ulinkpro/ulinkpro_cs_core_sight.htm
- [2] <http://elmicro.com/de/jtag-swd.html>